

### Aufgabe 1: Logische Uhren

- a) In einem verteilten System treten die Ereignisse *a* bis *l* auf. Die kausalen Abhängigkeiten sind durch das nachfolgende Happened-Before-Diagramm gegeben. Geben Sie für alle im Happened-Before-Diagramm vorhandenen Ereignisse jeweils die Lamportzeit und die Vektorzeit an.

Ereignis	Vektorzeit	Lamportzeit ( $P_1, P_2, P_3$ )
a	3	(1, 0, 2)
b	4	(2, 0, 2)
c	9	(3, 5, 3)
d	2	(0, 1, 1)
e	5	(2, 2, 2)
f	6	(2, 3, 2)
g	7	(2, 4, 3)
h	8	(2, 5, 2)
i	1	(0, 0, 1)
j	2	(0, 0, 2)
k	3	(0, 0, 3)
l	7	(2, 3, 4)

- b) Geben Sie an, welche der folgenden Behauptungen für die Lamportzeit zutrifft und welche der Behauptungen für die Vektorzeit zutrifft. Erläutern Sie weiterhin ihre Wahl.

Eine Lamport Uhr erfüllt nur die schwachen Konsistenzbedingung für Uhren, d.h. für die Lamportzeit gilt Aussage II), aber nicht I) und III). Man kann mithilfe der Lamportzeit also Ereignisse kausal nach dem Auftreten ordnen, aber keine Aussagen über Nebenläufigkeit und kausale Abhängigkeiten treffen.

Für die logische Uhr gelten alle drei Aussagen. Insbesondere erfüllt diese die starke Konsistenzbedingung für Uhren, d.h. I) und II) gelten beide. Da durch Vergleichen der Zeiten kausale Abhängigkeiten erkannt werden können, kann auch eine kausale Unabhängigkeit und damit Nebenläufigkeit erkannt werden. Somit gilt hier auch Aussage III).

### Aufgabe 3: Zeitsynchronisation

- a) Sei  $t$  der Zeitstempel des Servers und  $T_{round}$  die Zeitspanne von Anfrage bis Rückantwort. Dann ist:

$$T_{sync} = t + \frac{T_{round}}{2} = 10:14:05 + \frac{10:14:06 - 10:14:00}{2} = 10:14:05 + 00:00:03 = 10:14:08$$

- b) Die Berechnung nach Cristian beruht auf der Annahme, dass ein Paket für Hin- und Rückweg die gleiche Übertragungszeit hat. In obiger Betrachtung wurde zudem angenommen, dass die Bearbeitung der Anfrage und Erstellung einer Antwort keine Zeit beansprucht.

### Aufgabe 4: Bully-Algorithmus

- Der Bully-Algorithmus ist ein Auswahlalgorithmus der einen bestimmten Teilnehmer eindeutig bestimmt. In verteilten System kann hiermit z.B. beim Ausfall eines Koordinatorprozesses ein neuer Koordinator gewählt werden. Es wird hierbei angenommen, dass jeder Prozess die Id jedes anderen Prozesses im System kennt. Ein beliebiger Prozess, der den Ausfall des Koordinators registriert, z.B. durch ein Timeout, startet den Bully-Algorithmus indem er an alle Prozesse mit einer höheren Prozess-Id eine Wahlnachricht sendet. Empfängt ein Prozess eine Wahlnachricht, sendet dieser wiederum eine Wahlnachricht an alle Prozesse mit höherer Prozess-Id und eine Antwort an den Prozess, von dem die Wahlnachricht gesendet wurde.

Falls ein Prozess die höchste Prozess-Id besitzt oder keine Antwort von anderen Prozessen erhält, wird dieser der neue Koordinator.

Bully-Algorithmus für vier aktive Knoten mit den IDs (3, 5, 9, 11):

- (a)  $K_5$  beginnt und sendet Wahlnachricht an  $K_9$ ,  $K_{11}$ .
- (b)  $K_9$ ,  $K_{11}$  senden eine *alive*-Antwort an  $K_5$ .
- (c)  $K_9$  sendet Wahlnachricht an  $K_{11}$ .
- (d)  $K_{11}$  besitzt die höchste registrierte Prozess-Id und sendet seine eigene Wahl an die übrigen Prozesse.